ECHO REST Ingest Guide

July 2012

Date	Brief Description	Author
7/24/2012	Initial Revision	Katie Baynes

Overview

The purpose of this document is to guide new and existing ECHO data partners through the process of ingesting their data via our REST interface. It is intended to be a hands-on, step-by-step introduction to the process. Any comments or questions regarding the contents of this document should be directed to echo@echo.nasa.gov

If you are a first time REST provider, we strongly advise starting with extensive testing in our Testbed (Base URL: http://testbed.echo.nasa.gov/) or Partner Test environment (Base URL: http://api-test.echo.nasa.gov/)

Assumptions and Pre-requisites

This document is targeted at a technical audience and is intended to be a high-level roadmap for ingest implementation. While much of this document should be accessible to novice ECHO users, there are a few technologies and configurations that will prove useful to have in your tool belt to follow along with this document.

1. A working knowledge of REST concepts (http://en.wikipedia.org/wiki/Representational_state_transfer)
2. Familiarity with ECHO and the EOSDIS User Registration System (URS)
3. Familiarity with the ECHO10 format at both the dataset and granule level (Please note that adding and updating metadata via ISO 19115 is supported by our REST interface, but this document will utilize ECHO10 in its examples)
4. Ability to perform basic HTTP operations (GET, PUT, DELETE) via some sort of programmatic tool. This document will use curl in its examples, but there are several excellent command line (wget) and browser-based tools (REST Client for Firefox, Postman and other tools for Chrome)
5. A registered and active ECHO data provider configured for REST ingest. If you are unsure if you have provider configured for REST, please contact us. The following information is required:
    a. Provider Name (10 character limit)
    b. Organization name
    c. Provider Contact information
    d. Discovery URL (usually the URL of your organization's website)
    e. Description of Holdings
6. A registered and active URS account (http://urs.earthdata.nasa.gov) that has been designated as having administrative privileges for your provider via ECHO's PUMP interface. This may require coordination with us.

Document Conventions

REST call parameters are shown using the following format:

Route:	https://api-test.echo.nasa.gov/<route>
Verb:	{GET, PUT, POST, DELETE}
Header:	Name = Value

The following styling is used for sample xml bodies and example responses:

```
<element>
  <subelement>value</subelement>
  <subelement>value</subelement>
</element>
```

The following styling is used for sample command line curl invocations:

```
curl https://api-test.echo.nasa.gov
```

The following styling is used for tips and best practices:

Be sure to drink your Ovaltine!

Validating your Metadata

If you are a first time ECHO provider, your best course of action is to validate that your data to be ingested is valid with respect to the ECHO10 schema:

http://www.echo.nasa.gov/ingest/schemas/operations/docs/index.html

It is also **highly** recommended that you vet your data with the ECHO team by submitting a sample file. We often have recommendations that could help ensure your data is easy to find by your users.

If you are going to use ISO 19115 for ingest, please refer the following links:

ISO 19115 Home:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020

ISO 19115 Schema Currently Supported by ECHO:
http://www.isotc211.org/2005/

Earthdata ISO 19115 Home Wiki:
http://earthdata.nasa.gov/wiki/main/index.php/ISO_19115_Home

Ingesting your Data

Acquiring an ECHO token

The first step in your successful ingest is creating a token that has provided you with authentication and authorization for subsequent REST requests. The user used to create the token must be authorized via PUMP to ingest data. If you have problems, please first check that your account has the proper permissions within PUMP. When logging in to PUMP with this account, if you see a tab labeled "Provider Context" you should be ok to go. If you have problems, contact us.

Note that tokens are only valid for 30 days.

Method Outline:

Route:	https://api-test.echo.nasa.gov/echo-rest/tokens
Verb:	POST
Header:	Content-Type = application/xml

Sample XML Body:

```
<token>
      <username>sample_username</username>
      <password>sample_password</password>
      <client_id>client_name_of_your_choosing</client_id>
      <user_ip_address>your_origin_ip_address</user_ip_address>
      <provider>your_provider_id</provider>
</token>
```

Curl Example:

```
curl –X POST -d "<token><username>kermit</username>
<password>@t43Fr0G@</password><client_id>Muppet
Ingest</client_id><user_ip_address>127.0.0.1</user_ip_address>
<provider>MUPPET_DC</provider></token>" https://api-
test.echo.nasa.gov/echo-rest/tokens
```

Sample Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<token>
  <id>75E5CEBE-6BBB-2FB5-A613-0368A361D0B6</id>
  <username>sample_username</username>
  <client_id>client_name_of_your_choosing</client_id>
  <user_ip_address>your_origin_ip_address</user_ip_address>
  <provider>CDDIS</provider>
</token>
```

Adding a dataset

Here we get to the meat of the ingest process. This is the outline of dataset ingest. This is a synchronous operation and your data should be almost immediately available for retrieval.

Method Outline:

Route:	https://api-test.echo.nasa.gov/catalog-rest/providers/<provider id>/datasets/<dataset id>
Verb:	PUT
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Sample XML Body:

Note, order is important for xml elements. Please refer to the schemas linked in the "Validating your Metadata" section of this document. Also note that this example abridges some of the dataset metadata for the sake of brevity. The route table entry above lists <dataset_id>, this refers to the <DataSetId> element from the dataset. If you are going to be invoking this programmatically (i.e. not through a browser plug-in), please ensure that this element has been URL encoded; any spaces or other potential special characters will cause errors in your insert.

Note: For URL encoding of the dataset id, the following tool can help: ***http://meyerweb.com/eric/tools/dencoder/***

```
<Collection
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Collection.xsd">
  <ShortName>COLL_SHORT_NAME</ShortName>
  <VersionId>1</VersionId>
  <InsertTime>2012-05-15T10:00:00Z</InsertTime>
  <LastUpdate>2012-05-15T12:00:00Z</LastUpdate>
  <LongName>Full Resolution Calibrated TOA Radiance</LongName>
  <DataSetId>Full Resolution Calibrated TOA Radiance V1</DataSetId>
  ...
</Collection>
```

Curl Examples:

Again, note the abridged xml content. If you have a problem returning a **417** status code from your REST route invocation, include the "Expect: " header as shown in the snippet below.

```
curl -X PUT -H "Content-Type: application/xml" -H "Expect: " -H "Echo-
Token: 75E5CEBE-6BBB-2FB5-A613-0368A361D0B6" -d '<Collection
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:/Users/kbaynes/work/echo/echo-
ingest2/support/bindings/Collection.xsd">
<ShortName>MER_FR__1P</ShortName> <VersionId>1</VersionId>
<InsertTime>2012-05-15T10:00:00Z</InsertTime> <LastUpdate>2012-05-
15T12:00:00Z</LastUpdate> <LongName>Full Resolution Geolocated and
Calibrated TOA Radiance</LongName> <DataSetId>Full Resolution
Geolocated and Calibrated TOA Radiance V1</DataSetId> ... </Collection>
' https://api-test.echo.nasa.gov/catalog-
rest/providers/LAADS/datasets/Full%20Resolution%20Geolocated%20and%20Ca
librated%20TOA%20Radiance%20V1
```

For curl users, if your xml is in a file, use the "@" symbol to prefix the file name. For example, In this case the file name is "myDataset.xml"

```
curl -X PUT -H "Content-Type: application/xml" -H "Expect: " -H "Echo-Token: 75E5CEBE-6BBB-2FB5-A613-0368A361D0B6" –d @myDataset.xml https://api-test.echo.nasa.gov/catalog-rest/providers/LAADS/datasets/Full%20Resolution%20Geolocated%20and%20Calibrated%20TOA%20Radiance%20V1
```

Retrieving a dataset

Once your dataset has been ingested via the RESTful PUT operation above, you can verify that the data can be retrieved. The body of the http result will contain your dataset

Method Outline:

Route:	https://api-test.echo.nasa.gov/catalog-rest/providers/<provider id>/datasets/<dataset id>
Verb:	GET
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Curl Example:

```
curl -X GET -H "Content-Type: application/xml" -H "Echo-Token: 75E5CEBE-6BBB-2FB5-A613-0368A361D0B6" https://api-test.echo.nasa.gov/catalog-rest/providers/LAADS/datasets/Full%20Resolution%20Geolocated%20and%20Calibrated%20TOA%20Radiance%20V1
```

Updating a dataset

The update of a dataset is done in the exact manner as an initial ingest. You may follow the same instruction set outlined in "Ingesting a Dataset"

Deleting a dataset

We recommend only doing a DELETE if you are **very** confident in your action and in your ability to recover from such an action. This action will delete the dataset and **ALL** granules that refer to this dataset.

Method Outline:

Route:	https://api-test.echo.nasa.gov/catalog-rest/providers/<provider id>/datasets/<dataset id>
Verb:	DELETE
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Curl Example:

```
curl -X DELETE -H "Content-Type: application/xml" -H "Echo-Token: 75E5CEBE-6BBB-2FB5-A613-0368A361D0B6" https://api-test.echo.nasa.gov/catalog-rest/providers/LAADS/datasets/Full%20Resolution%20Geolocated%20and%20Calibrated%20TOA%20Radiance%20V1
```

Adding a granule

Adding a granule to an existing dataset is quite similar to adding the dataset itself.

Method Outline:

Route:	https://api-test.echo.nasa.gov/catalog-rest/providers/<provider id>/granules/<granule_ur >
Verb:	PUT
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Sample XML Body:

Note, order is important for xml elements. Please refer to the schemas linked in the "Validating your Metadata" section of this document. Also note that this example abridges some of the granule metadata for the sake of brevity. The route table entry above lists <granule_ur>; this refers to the <GranuleUR> element from the granule. The <Collection> element should include a reference to the dataset id of the parent dataset.

```
<Granule xmlns:html="http://www.w3.org/1999/xhtml"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="Granule.xsd">
    <GranuleUR>GranuleUR0</GranuleUR>
    <InsertTime>2006-05-04T18:13:51.0Z</InsertTime>
    <LastUpdate>2006-05-04T18:13:51.0Z</LastUpdate>
    <DeleteTime>2006-05-04T18:13:51.0Z</DeleteTime>
    <Collection>
        <DataSetId>DataSetId0</DataSetId>
    </Collection>
    <RestrictionFlag>0</RestrictionFlag>
    <RestrictionComment>RestrictionComment0</RestrictionComment>
    <DataGranule>
        <SizeMBDataGranule>0</SizeMBDataGranule>
        <ReprocessingPlanned>ReprocessingPlanned0</ReprocessingPlanned>
        <ReprocessingActual>ReprocessingActual0</ReprocessingActual>
        <ProducerGranuleId>ProducerGranuleId0</ProducerGranuleId>
        <DayNightFlag>DAY</DayNightFlag>
        <ProductionDateTime>2006-05-04T18:13:51.0Z</ProductionDateTime>
        <LocalVersionId>LocalVersionId0</LocalVersionId>
    </DataGranule>
    ...
    <OnlineAccessURLs>
        <OnlineAccessURL>
            <URL>URL0</URL>
            <URLDescription>URLDescription0</URLDescription>
            <MimeType>MimeType0</MimeType>
        </OnlineAccessURL>
    </OnlineAccessURLs>
    ...
    <Orderable>false</Orderable>
    <DataFormat>DataFormat0</DataFormat>
    <Visible>false</Visible>
    <CloudCover>0</CloudCover>
    ...
</Granule>
```

Curl Example:

Similar to dataset ingest, for curl users, if your xml is in a file, use the "@" symbol to prefix the file name. For example, In this case the file name is "myGranule.xml"

```
curl -X PUT -H "Content-Type: application/xml" -H "Expect: " -H "Echo-Token: 75E5CEBE-6BBB-2FB5-A613-0368A361D0B6" –d @myGranule.xml https://api-test.echo.nasa.gov/catalog-rest/providers/LAADS/granules/GranuleUR0
```

Retrieving a granule

Once your granule has been ingested via the RESTful PUT operation above, you can verify that the data can be retrieved. The body of the http result will contain your granule in ECHO10 format.

Method Outline:

Route:	https://api-test.echo.nasa.gov/catalog-rest/providers/<provider id>/granules/<granule_ur >
Verb:	GET
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Updating a granule

The update of a granule is done exact manner as an initial ingest. You may follow the same instruction set outlined in "Ingesting a Granule"

Deleting a granule

We recommend only doing a DELETE if you are **very** confident in your action and in your ability to recover from such an action.

Method Outline:

Route:	https://api-test.echo.nasa.gov/catalog-rest/providers/<provider id>/granules/<granule_ur>
Verb:	DELETE
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Curl Example:

```
curl -X DELETE -H "Content-Type: application/xml" -H "Echo-Token: 75E5CEBE-6BBB-2FB5-A613-0368A361D0B6" https://api-test.echo.nasa.gov/catalog-rest/providers/LAADS/granules/GranuleUR0
```

Removing your token

The final method this document covers involves cleaning up a token when you are finished with it. If you are performing programmatic ingest and want to generate a new token for each batch of inserts you are doing, we recommend this best practice of deleting your token. This will invalidate your token.

Method Outline:

Route:	https://api-test.echo.nasa.gov/echo-rest/tokens/75E5CEBE-6BBB-2FB5-A613-0368A361D0B6
Verb:	DELETE
Header:	Content-Type = application/xml
Header:	Echo-Token = <token created above>

Full API Documentation

For a dictionary of methods available via our REST interfaces, please refer to the following links.

https://api-test.echo.nasa.gov/catalog-rest/
https://api-test.echo.nasa.gov/echo-rest/

Troubleshooting Guide

Question

I keep getting an error similar to the following when I try to ingest/retrieve my data:

```
<errors><error>Do not have UPDATE/INGEST ingest management permission
on provider PROVIDER.</error></errors>
```

Answer

You need to include a token with the proper permissions. Check and make sure you have included the <provider> element when you generate your token. Also, you should log in to **PUMP** to ensure that the account you are using to generate your token has the appropriate permissions. Check the following link for the appropriate URL for your environment: https://earthdata.nasa.gov/about-eosdis/system-description/about-echo/echo-systems

Questions To Be Added as Needed